Ignacio Martín Franco

[Dirección de correo electrónico]

Autores

Ignacio Martín Franco UO239718 71727308Z  
Grabriela ……………………………………………………..

DOcuMENTACIÓN  
Red social

Sdi1-UO231379UO239718

Índice

[Una descripción clara y detallada de la implementación de cada uno de los casos de uso implementados. 2](#_Toc508983314)

[1 Público: registrarse como usuario 2](#_Toc508983315)

[2 Público: iniciar sesión 2](#_Toc508983316)

[3 Usuario registrado: listar todos los usuarios de la aplicación 2](#_Toc508983317)

[4 Usuario registrado: buscar entre todos los usuarios de la aplicación 2](#_Toc508983318)

[5 Usuario registrado: enviar una invitación de amistad a un usuario 3](#_Toc508983319)

[6 Usuario registrado: listar las invitaciones de amistad recibidas 3](#_Toc508983320)

[7 Usuario registrado: aceptar una invitación recibida 4](#_Toc508983321)

[8 Usuario registrado: listar los usuarios amigos 4](#_Toc508983322)

[9 Usuario registrado: crear una nueva publicación 6](#_Toc508983323)

[10 Usuario registrado: listar mis publicaciones 6](#_Toc508983324)

[Un catálogo de las pruebas unitarias realizadas y descripción sencilla de cada una 7](#_Toc508983325)

[1.1 [RegVal] 7](#_Toc508983326)

[1.2 [RegInval] 7](#_Toc508983327)

[2.1 [InVal] 7](#_Toc508983328)

[2.2 [InInVal] 7](#_Toc508983329)

[3.1 [LisUsrVal] 7](#_Toc508983330)

[Acceso al listado de usuarios desde un usuario en sesión 7](#_Toc508983331)

[3.2 [LisUsrInVal] 7](#_Toc508983332)

[4.1 [BusUsrVal] 7](#_Toc508983333)

[4.2 [BusUsrInVal] 7](#_Toc508983334)

[5.1 [InvVal] 7](#_Toc508983335)

[5.2 [InvInVal] 7](#_Toc508983336)

[6.1 [LisInvVal] 7](#_Toc508983337)

[7.1 [AcepInvVal] 8](#_Toc508983338)

[8.1 [ListAmiVal] 8](#_Toc508983339)

[9.1 [PubVal] 8](#_Toc508983340)

[10.1 [LisPubVal] 8](#_Toc508983341)

[11.1 [LisPubAmiVal] 8](#_Toc508983342)

[11.2 [LisPubAmiInVal] 8](#_Toc508983343)

[12.1 [PubFot1Val] 8](#_Toc508983344)

[12.1 [PubFot2Val] 8](#_Toc508983345)

[13.1 [AdInVal] 8](#_Toc508983346)

[13.2 [AdInInVal] 8](#_Toc508983347)

[14.1 [AdLisUsrVal] 8](#_Toc508983348)

[15.1 [AdBorUsrVal] 8](#_Toc508983349)

[15.2 [AdBorUsrInVal] 8](#_Toc508983350)

[Cualquier otra información necesaria para una descripción razonablemente detallada de lo entregado y su correcto despliegue y ejecución. 9](#_Toc508983351)

# Una descripción clara y detallada de la implementación de cada uno de los casos de uso implementados.

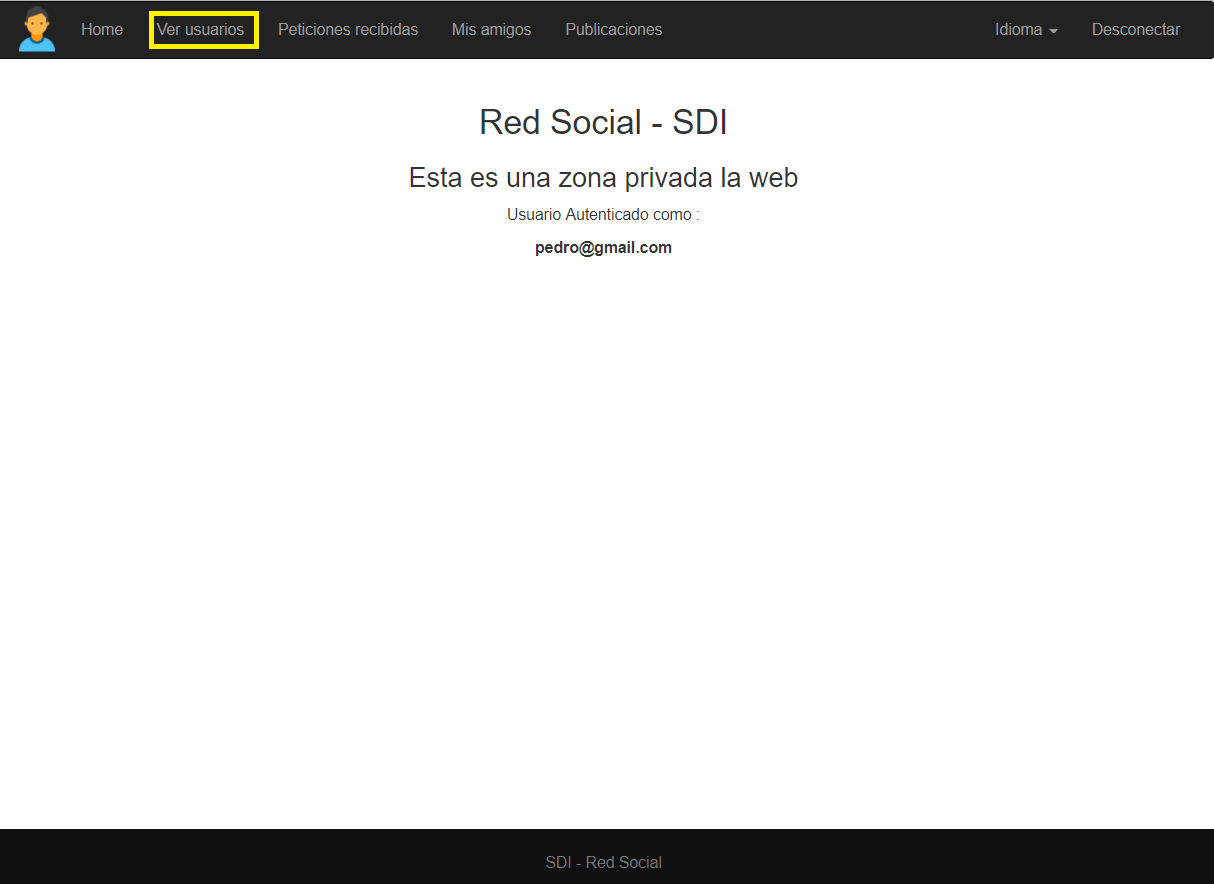
## 1 Público: registrarse como usuario

## 2 Público: iniciar sesión

## 3 Usuario registrado: listar todos los usuarios de la aplicación

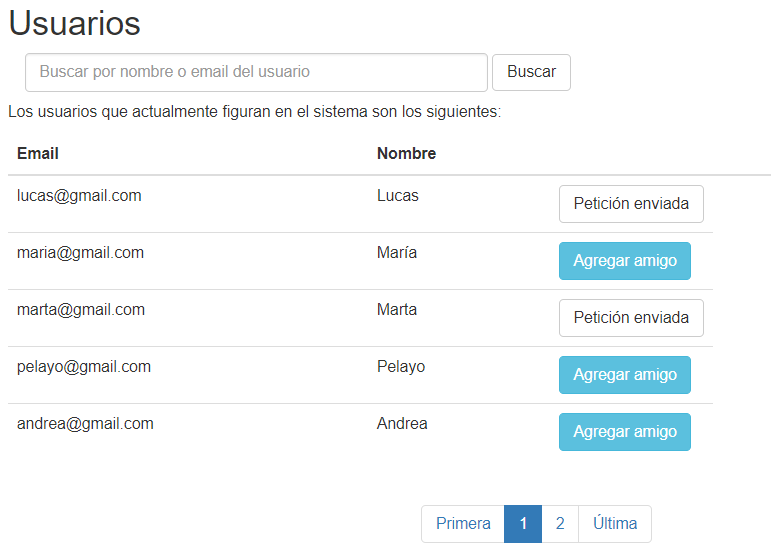
## 4 Usuario registrado: buscar entre todos los usuarios de la aplicación

Una vez registrado el usuario válido (en este caso [pedro@gmail.com](mailto:pedro@gmail.com) con contraseña 123456) se hace click sobre “Ver usuarios”:



A continuación, se mostrará una lista los usuarios mostrando nombre y email junto con un botón que tendrá 3 posibles valores:

* “Amigo” si el usuario es amigo del usuario en sesión
* “Agregar amigo”: al hacer click mandará una petición de amistad a dicho usuario
* “Petición enviada”: si se ha mandado una petición y está pendiente de ser aceptada



## 5 Usuario registrado: enviar una invitación de amistad a un usuario

Una vez seguidos los pasos del caso de uso anterior (4 Usuario registrado: buscar entre todos los usuarios de la aplicación) se podrá enviar una invitación de amistad a aquellos usuarios que tengan asociados el botón con el mensaje “Agregar amigo”. Al hacer click el valor del botón pasará a ser “Petición enviada”.



Además, se mandará una petición de amistad al usuario asociado al botón pulsado

## 6 Usuario registrado: listar las invitaciones de amistad recibidas

Una vez registrado el usuario válido (en este caso [pedro@gmail.com](mailto:pedro@gmail.com) con contraseña 123456) se hace click sobre “Peticiones recibidas”:



A continuación, se mostrará una lista de las peticiones recibidas que tiene el usuario.



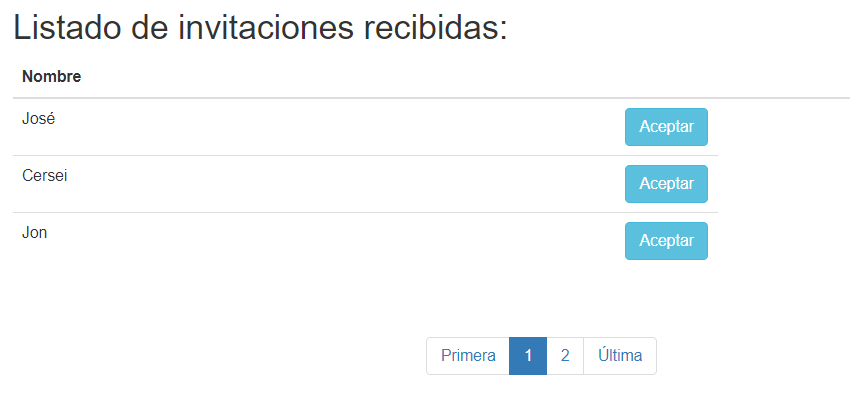
## 7 Usuario registrado: aceptar una invitación recibida

Siguiendo el caso de uso anterior (6 Usuario registrado: listar las invitaciones de amistad recibidas).

Al hacer click en el botón “Aceptar” de algún usuario:

1. Se añaden mutuamente como amigos a sus listas de amigos
2. Se elimina petición de la lista del usuario en sesión
3. Se elimina la petición y también si existía la misma petición, pero en dirección inversa (usuario en sesión mandó petición de amistad)

Tras aceptar a “Andrea”, “María” y “Sansa”



## 8 Usuario registrado: listar los usuarios amigos

Una vez registrado el usuario válido (en este caso [pedro@gmail.com](mailto:pedro@gmail.com) con contraseña 123456) se hace click sobre “Ver usuarios” y comprobamos que no tiene amigos:



Tras ir a “Invitaciones recibidas” y aceptar todas las invitaciones recibidas (de “Andrea”, “María”, “José”, “Sansa”, “Cersei” y “Jon”)



Y en la vista “Mis amigos”:



Si cerramos sesión haciendo click en “Desconectar” e iniciamos sesión con [andrea@gmail.com](mailto:andrea@gmail.com) comprobaremos que tiene como amigo a “Pedro”:





Y en el link “Mis amigos” mostrará como amigo a “Pedro”:



## 9 Usuario registrado: crear una nueva publicación

Se debe incluir una nueva vista con un formulario para que un usuario en sesión pueda crear una nueva publicación. Las publicaciones consistirán en un título y un texto, al crear la publicación se debe almacenar: que usuario realizo la publicación, la fecha en la que se creó, el título y el texto asociados a la publicación.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al formulario de crear publicaciones.

## 10 Usuario registrado: listar mis publicaciones

Se visualizará en una lista con todas las publicaciones realizadas por el usuario en sesión. Para cada publicación se mostrará: la fecha en la que se creó y el título de la publicación.

No es necesario incluir sistema de paginación en este listado.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al a la lista de publicaciones de un usuario.

# Un catálogo de las pruebas unitarias realizadas y descripción sencilla de cada una

## [RegVal]

Registro de Usuario con datos válidos.

## 1.2 [RegInval]

Registro de Usuario con datos inválidos (repetición de contraseña invalida).

## 2.1 [InVal]

Inicio de sesión con datos válidos.

## 2.2 [InInVal]

Inicio de sesión con datos inválidos (usuario no existente en la aplicación).

3.1 [LisUsrVal]

Acceso al listado de usuarios desde un usuario en sesión.

## 3.2 [LisUsrInVal]

Intento de acceso con URL desde un usuario no identificado al listado de usuarios desde un usuario en sesión. Debe producirse un acceso no permitido a vistas privadas.

## 4.1 [BusUsrVal]

Realizar una búsqueda valida en el listado de usuarios desde un usuario en sesión.

## 4.2 [BusUsrInVal]

Intento de acceso con URL a la búsqueda de usuarios desde un usuario no identificado. Debe producirse un acceso no permitido a vistas privadas.

## 5.1 [InvVal]

Enviar una invitación de amistad a un usuario de forma valida.

1. **Nos registramos coomo usuario “Pedro”, email: pedro@gmail.com**

PO\_HomeView.*clickOption*(*driver*, "login", "id", "btnIdentificate");

PO\_LoginView.*fillForm*(*driver*, "pedro@gmail.com", "123456");

1. **Vamos a la vista que lista todos los usuarios del sistema**

PO\_HomeView.*clickOption*(*driver*, "/user/list", "text", "Usuarios");

1. **Comprobamos que en los 5 primeros usuarios son Lucas, María, Marta, Pelayo y Andrea**

SeleniumUtils.*textoPresentePagina*(*driver*, "Lucas");

SeleniumUtils.*textoPresentePagina*(*driver*, "María");

SeleniumUtils.*textoPresentePagina*(*driver*, "Marta");

SeleniumUtils.*textoPresentePagina*(*driver*, "Pelayo");

SeleniumUtils.*textoPresentePagina*(*driver*, "Andrea");

1. **Click sobre botón de Pelayo con id=** resendButton13 **con mensaje “Agregar amigo”, al hacer click crea una petición hecha por Pedro con destinatario Pelayo. El texto del botón pasa a ser “Petición enviada”**

*assertEquals*("Agregar amigo",PO\_NavView.*getButtonText*(*driver*, "resendButton13"));

PO\_NavView.*clickButton*(*driver*, "resendButton13");

## 5.2 [InvInVal]

Enviar una invitación de amistad a un usuario al que ya le habíamos invitado la invitación previamente. No debería dejarnos enviar la invitación, se podría ocultar el botón de enviar invitación o notificar que ya había sido enviada previamente.

1. **Nos registramos coomo usuario “Pedro”, email: pedro@gmail.com**

PO\_HomeView.*clickOption*(*driver*, "login", "id", "btnIdentificate");

PO\_LoginView.*fillForm*(*driver*, "pedro@gmail.com", "123456");

1. **Vamos a la vista que lista todos los usuarios del sistema**

PO\_HomeView.*clickOption*(*driver*, "/user/list", "text", "Usuarios");

1. **Comprobamos que en los 5 primeros usuarios son Lucas, María, Marta, Pelayo y Andrea**

SeleniumUtils.*textoPresentePagina*(*driver*, "Lucas");

SeleniumUtils.*textoPresentePagina*(*driver*, "María");

SeleniumUtils.*textoPresentePagina*(*driver*, "Marta");

SeleniumUtils.*textoPresentePagina*(*driver*, "Pelayo");

SeleniumUtils.*textoPresentePagina*(*driver*, "Andrea");

1. **Click sobre botón de Lucas con id=noresendButton4, como ya le hemos enviado una petición y no ha contestado, el botón tiene el texto “Petición enviada” y no tiene ninguna funcionalidad**

*assertEquals*("Petición enviada",PO\_NavView.*getButtonText*(*driver*, "noresendButton4"));

PO\_NavView.*clickButton*(*driver*, "noresendButton4");

1. **El botón sigue igual**

*assertEquals*("Petición enviada",PO\_NavView.*getButtonText*(*driver*, "noresendButton4"));

## 6.1 [LisInvVal]

Listar las invitaciones recibidas por un usuario, realizar la comprobación con una lista que al menos tenga una invitación recibida.

1. **Nos registramos coomo usuario “Pedro”, email: pedro@gmail.com**

PO\_HomeView.*clickOption*(*driver*, "login", "id", "btnIdentificate");

PO\_LoginView.*fillForm*(*driver*, "pedro@gmail.com", "123456");

1. **Vamos a la vista que lista las invitaciones recibidas**

PO\_HomeView.*clickOption*(*driver*, "/request/list", "text","Listado de invitaciones recibidas:");

1. **Comprobamos que los usuarios que han mandado invitaciones a Pedro son Andrea, María, José, Sansa y Cersei**

SeleniumUtils.*textoPresentePagina*(*driver*, "María");

SeleniumUtils.*textoPresentePagina*(*driver*, "José");

SeleniumUtils.*textoPresentePagina*(*driver*, "Sansa");

SeleniumUtils.*textoPresentePagina*(*driver*, "Cersei");

SeleniumUtils.*textoPresentePagina*(*driver*, "Jon");

## 7.1 [AcepInvVal]

Aceptar una invitación recibida.

1. **Nos registramos como usuario “Pedro”, email: pedro@gmail.com**

PO\_HomeView.*clickOption*(*driver*, "login", "id", "btnIdentificate");

PO\_LoginView.*fillForm*(*driver*, "pedro@gmail.com", "123456");

1. **Vamos a la vista que lista las invitaciones recibidas**

PO\_NavView.*clickOption*(*driver*, "/request/list", "text", "Listado de invitaciones recibidas:");

1. **Comprobamos que los usuarios que han mandado invitaciones a Pedro son Andrea, María, José, Sansa y Cersei**

SeleniumUtils.*textoPresentePagina*(*driver*, "Andrea");

SeleniumUtils.*textoPresentePagina*(*driver*, "María");

SeleniumUtils.*textoPresentePagina*(*driver*, "José");

SeleniumUtils.*textoPresentePagina*(*driver*, "Sansa");

SeleniumUtils.*textoPresentePagina*(*driver*, "Cersei");

1. **Aceptamos invitación de Andrea**

PO\_NavView.*clickButton*(*driver*, "acceptFriendButton24");

SeleniumUtils.*esperarSegundos*(*driver*, 2);

1. **Comprobamos que la invitación de Andrea ya no está**

SeleniumUtils.*textoNoPresentePagina*(*driver*, "Andrea");

## 8.1 [ListAmiVal]

Listar los amigos de un usuario, realizar la comprobación con una lista que al menos tenga un amigo.

## 9.1 [PubVal]

Crear una publicación con datos válidos.

## 10.1 [LisPubVal]

Acceso al listado de publicaciones desde un usuario en sesión.

## 11.1 [LisPubAmiVal]

Listar las publicaciones de un usuario amigo

## 11.2 [LisPubAmiInVal]

Utilizando un acceso vía URL tratar de listar las publicaciones de un usuario que no sea amigo del usuario identificado en sesión.

## 12.1 [PubFot1Val]

Crear una publicación con datos válidos y una foto adjunta.

## 12.1 [PubFot2Val]

Crear una publicación con datos válidos y sin una foto adjunta.

## 13.1 [AdInVal]

Inicio de sesión como administrador con datos válidos.

## 13.2 [AdInInVal]

Inicio de sesión como administrador con datos inválidos (usar los datos de un usuario que no tenga perfil administrador).

## 14.1 [AdLisUsrVal]

Desde un usuario identificado en sesión como administrador listar a todos los usuarios de la aplicación.

## 15.1 [AdBorUsrVal]

Desde un usuario identificado en sesión como administrador eliminar un usuario existente en la aplicación.

## 15.2 [AdBorUsrInVal]

Intento de acceso vía URL al borrado de un usuario existente en la aplicación. Debe utilizarse un usuario identificado en sesión pero que no tenga perfil de administrador.

# Cualquier otra información necesaria para una descripción razonablemente detallada de lo entregado y su correcto despliegue y ejecución.